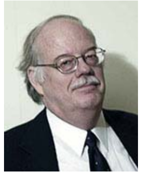
A Survey of Cognitive Architectures



David E. Kieras

University of Michigan

Powered by Naval Research TM/SM

Introduction to the Session

This session presents a survey of the cognitive architectures that are “out there” that might be suitable for user interface design purposes.

Each is summarized and briefly critiqued.

Note that the opinions expressed about this are mine - others may well disagree.

Also, systems under current development are changing rapidly, so their properties can change overnight.

Final summary is in terms of how well important constraints are represented and enforced.

Introduction

Approach

Desirable Properties of a Cognitive Architecture for User Interface Models

Desirable Properties (continued)

Other Desiderata

Modeling without Psychological Constraints

Modeling without a Learning Mechanism

Critical Constraints

Non-Critical Constraints

Approach

Many architectures and modeling systems out there.

- Most have little psychological content.
  - This is very hard to do.
- Some have "kitchen sink" character.
  - Every concept is thrown in - doesn't help!
- Many are actually general-purpose simulation systems.
  - May be useful on other grounds, but not necessarily for human performance modeling.

Criteria for Coverage Here

- Must be a computational simulation system.
- Must be primarily oriented to modeling human performance.
- Must be current or available.

Pictures, diagrams shown are from various project web sites.

Desirable Properties of a Cognitive Architecture for User Interface Models

Explicit representation of declarative and procedural knowledge.

- Task strategies in an inspectable, programmable form.
- Provides direct linkage from task analysis results.
- Can be related directly to user interface procedures.

Account for performance at the task level.

- Must address overt behavior over extended time.
- Computer systems are designed to support jobs, not to exercise individual psychological mechanisms.

Relevant to design decisions.

- Predict, explain differences in usability and performance for:
  - I/O devices.
  - Screen contents and layout.
  - Menu and command organization and wording.
  - Operating procedures and training materials.

Accessible, current scientific basis.

- Best test is presence in the peer-reviewed scientific literature.

Desirable Properties (continued)

Useful at all points in the design process:

Initial top-down design from a task analysis.
Evaluation of design based on specifications alone.
Comparison of design to usability requirements.
Comparison of alternative designs.
Evaluation of prototype or final implemented designs.

Quantitative predictions.

Must supply "engineering-style" metrics.
Allow design tradeoffs to be analyzed and understood.
Vague, qualitative analyses have little value.

Support useful approximation.

Applicability is more important than *exact* results.
Heuristics for dealing with validity of approximation.

In the "sweet spot:"

Enough detail and accuracy to be useful in design.
Simple enough to learn and use in the time available.

Representation and enforcement of important constraints.

Modeling without Psychological Constraints

Can a modeling system be useful even if no psychological constraints?

Yes, if goal is to determine the logical adequacy of a strategy for a task.

Can represent plans, procedures, strategies, and run simulations of them.
Is correct behavior produced?
Are there time constraints on adequate performance?

If non-psychological simulation shows strategy is wrong, then you know that a human using the strategy will also be wrong!

Can serve as a way to explore domain knowledge and accuracy of representation.

Modeling without a Learning Mechanism

Some architectures learn how to behave. Is this essential?

Bias in AI and psychology towards learning as critical.

- AI: Learning as a way to avoid knowledge engineering.
- Psychology: Learning is the most distinctive human trait.

In design, not really concerned about how human learning works.

- Most issues: performance *after* system has been learned.
- Learnability can (and has) been predicted without a model that learns.
  - How much knowledge is required in a performance model.
  - E.g. count production rules.

Critical Constraints

Well-grounded psychological constraints:

- Perceptual-motor are best documented, most pervasive, and strongest.
- Working memory constraints are definitely present, but not necessarily well understood.
- Controversial psychological constraints can skew models.
  - E.g. traditional central bottleneck hypothesis.

Well-justified representational constraints:

- Should use knowledge representations that have a track record in relevant cognitive science fields, such as:
  - Capturing empirical effects in human performance.
  - Providing elegant and coherent computational systems.
  - E.g. production systems, propositional networks, schemas.
- Why use something novel or unusual unless scientifically established?

Non-Critical Constraints

Speculative quasi-psychological constraints:

Mechanisms that constrain form of models, but no substantial scientific psychological basis for the particular mechanisms.

E.g. task scheduling using a algorithm or mechanism adapted from computer science or AI.

Implementation and application constraints:

Missing functionality or coverage - doesn’t undermine what is present, but may disqualify the system for a particular project.

Pragmatic issues: programming language used; training and support, licensing.

No conceptual or scientific content, but can make tremendous practical difference.

The Architectures

EPIC
Soar, EPIC-Soar
Soar, EPIC-Soar (continued)
ACT-R, ACT-R/PM
ACT-R, ACT-R/PM (continued)
ACT-R, ACT-R/PM (continued)
GLEAN
APEX/CPM-GOMS
MIDAS
MIDAS (continued)
MIDAS (continued)
MicroSAINT IPME/MS-HOS
MicroSAINT IPME/MS-HOS (continued)
COGNET/iGen

EPIC

Urls:

http://www.eecs.umich.edu/~kieras/epic.html
http://www.umich.edu/~bcalab/epic.html

A production system cognitive processor surrounded by perceptual and motor processors.

Originally developed to model multimodal multiple task situations.

Programmed by writing productions, setting some perceptual/motor parameters.

Pros

Production system, most adequate type of architecture at this time.

Relatively simple to program due to underlying architectural simplicity.

Good support for modeling multiple complex tasks and perceptual/motor issues.

Cons

Primarily an academic research system; limited support.

No coverage of intermediate to longer-term memory phenomena.

Many details & parameters of perceptual/motor systems.

- Modeler may have to confront more details than really wants to.

Soar, EPIC-Soar

Url:

http://sitemaker.umich.edu/soar

Successful and powerful production system.

Can represent extremely large and complex rule sets.

Developed by Newell, Laird, & Rosenbloom, closely related to general GOMS theory.

Primary use in AI, but also in cognitive modeling; has been combined with EPIC's perceptual-motor processors.

Production system has a three-stage process:

Rules propose operators that should help solve the current problem.

Preferences then apply to select best operator.

Operator is applied to change the state of the external world and production system.

Heavy emphasis on learning in a problem-solving context.

New rules are created to short-cut lengthy sequences of rule/operator processing - "chunking."

Soar, EPIC-Soar (continued)

Unusual feature: No declarative knowledge representation.

Pros

Well established in AI community, good software support and training.
Commercial-grade applications developed and sold.
Commercial vendor of Soar-based systems:
http://www.soartech.com
Soar itself is freely available.
Definite track record of success on large problems (thousands of rules).

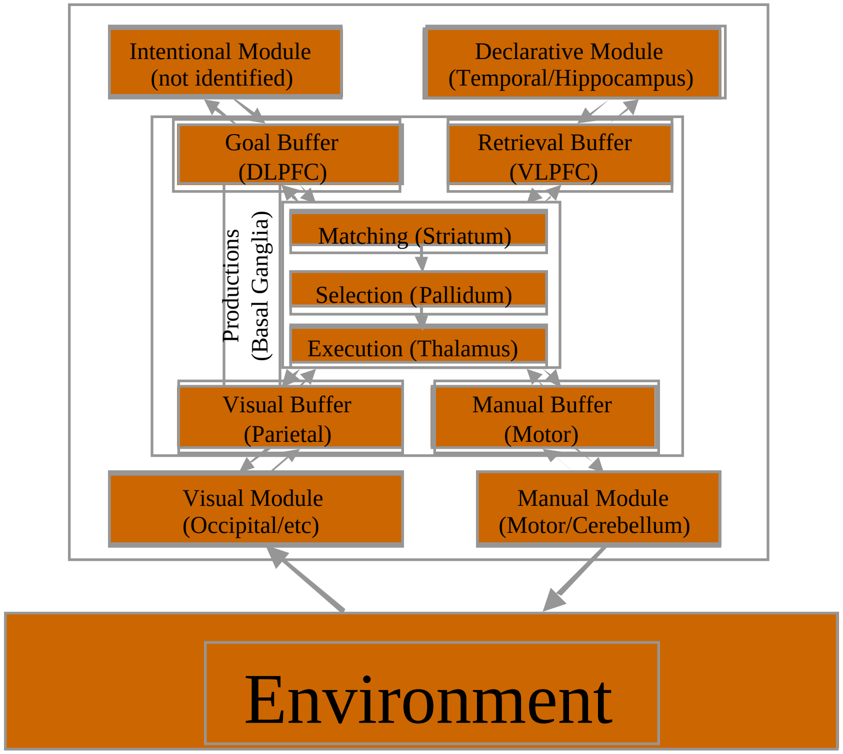
Cons

Psychological basis of learning and reasoning not as well developed as others.
Reputed to be difficult to program, even with excellent software support.

ACT-R, ACT-R/PM

Url:
http://act-r.psy.cmu.edu/

A “hybrid” production system architecture.
Neural-net like activation mechanism controls rules.
Architecture related to components from cognitive neuroscience.
Heavy emphasis on modeling learning.



ACT-R, ACT-R/PM (continued)

A production system processor surrounded by perceptual and motor systems.

- Simple visual system.
- Motor systems adopted from EPIC.
- Peripherals interface to production rule cognitive processor through limited-capacity “buffers”.
- Major changes every few years.
  - Current version (5) supersedes previous ACT-R/PM.

Heavy emphasis on activation and learning mechanisms.

- New rules constructed as a result of prior activity.
  - (Different versions follow different principles.)
- Production rule is triggered if condition items are adequately activated.
- Rules are strengthened by prior success - a learning mechanism.

Programmed by writing productions, setting some activation and perceptual/motor parameters.

ACT-R, ACT-R/PM (continued)

Pros

Production system, most adequate type of architecture at this time.
Very active user community, primarily psychologists.
Strong training, support, and commitment to “user friendly” software.
Best known of current psychology-based systems.

Cons

Primarily academic research system; value for large applied problems not clear.
Scientific basis can be debated (IMO).
Perceptual/motor representations lagging.
Many parameters that might affect performance (cf. EPIC).
Not sure what practical modeler would do.

GLEAN

Url:

ftp://www.eecs.umich.edu/people/kieras/GOMS

A simplified cognitive architecture for GOMS modeling.

Keystroke-Level Model grain in perceptual/motor operators.

Model programmed with a high-level GOMS-based language.

Statements correspond to production rules.

Pros

Relatively easily and simple to learn and use.
Fairly complete GOMS implementation, well-documented.
Extremely portable and robust software.
Favorable experience with use by novice modelers.

Cons

Clumsy interface due to software portability restrictions.
Currently not commercial-grade; Limited training and support available.
KLM accuracy and grain size may not be good enough.

APEX/CPM-GOMS

Url:
http://www.andrew.cmu.edu/~bj07/apex/

Automates construction of CPM-GOMS models.
Resource allocation by APEX, a general-purpose simulation engine.

Programmed with an GOMSL-like language.
Includes syntax for specifying concurrency.

Pros
May allow a successful form of modeling to be done much more easily.
Might easily provide the “fastest possible” model for a task.

Cons
A research system, still under development.
Modeling language has intrusions from general-purpose simulation engine.
Psychological mechanisms limited to those in CPM-GOMS.

MIDAS

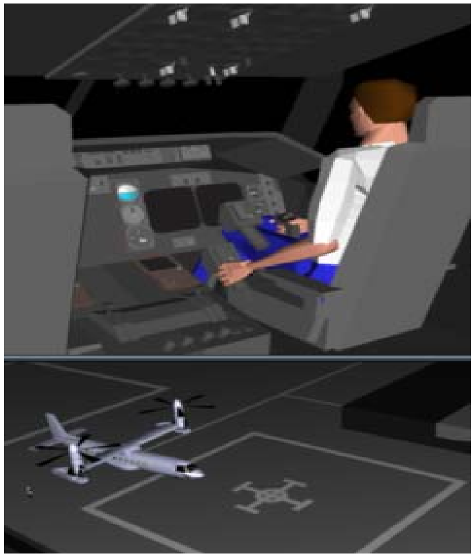
Urls:

http://caffeine.arc.nasa.gov/midas/
http://www-midas.arc.nasa.gov/
http://www.engr.sjsu.edu/hfe/hail/

Original domain was helicopter cockpit design.

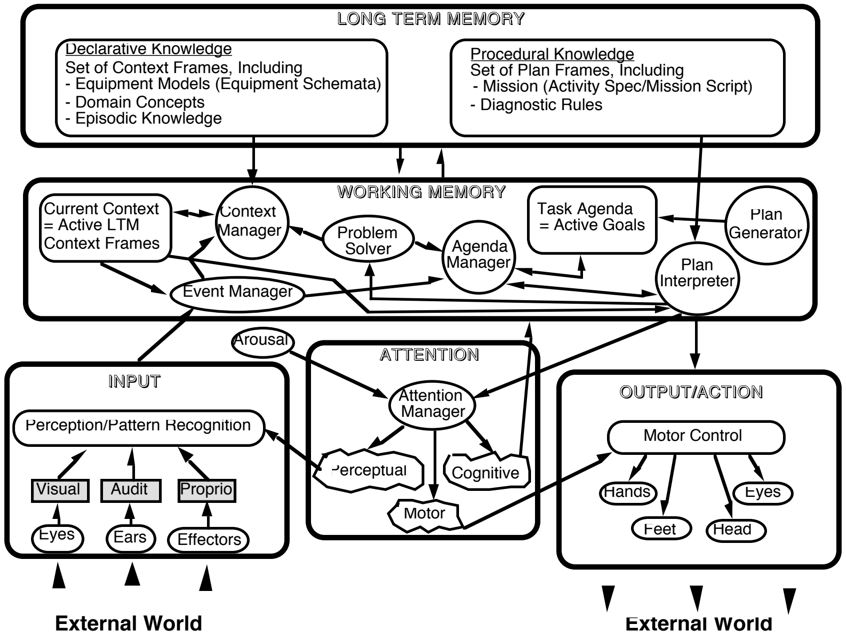
Including anthropometrics, physical layout, mission profile workload.

Goal is complete integration and visualization of design domain



MIDAS (continued)

A large and complicated system, whose psychological rationale isn't very well documented.



Papers describe a few basic psychological aspects, such as an elementary visual model.

Not clear how it is programmed.

MIDAS (continued)

Pros

Tackles large-scale design problems head-on - all aspects of design issues represented.

Well developed for aviation problems, NASA-based.

Cons

Not well documented; possibly over-complicated.

Current state is unclear - new design underway for some time now.

MicroSAINT IPME/MS-HOS

Url:

http://www.maad.com/

A task network system for human performance modeling.

- Includes an environmental layout representation and editing tool.
- Includes psychological micro-models.
  - MicroSAINT HOS implementation.

Pros

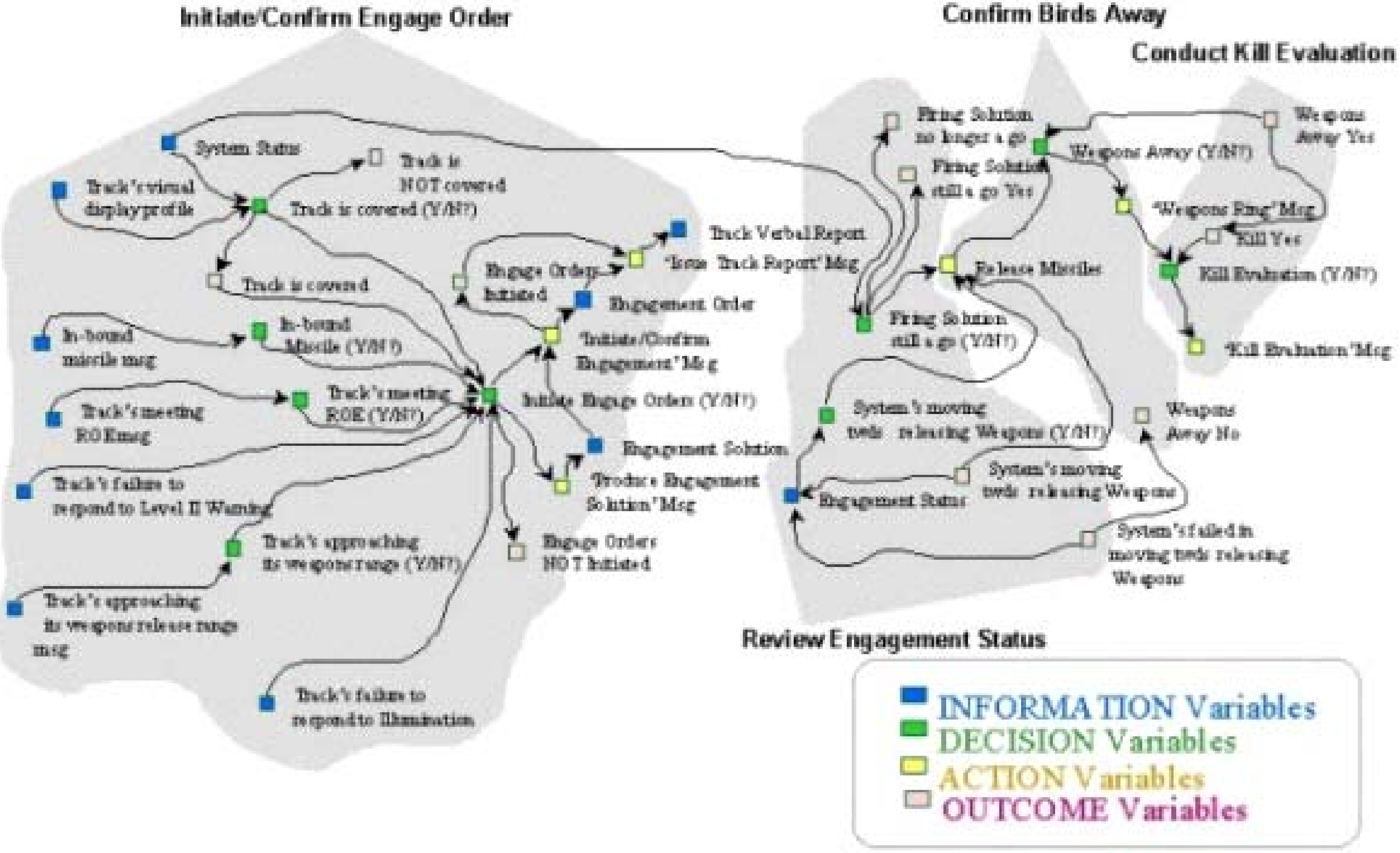
- Generality of modeling system means anything can be modeled from earliest stages of design, to detailed design.
- Well-established and well-known methodology and software tools - a defacto standard.
- Commercial product; excellent training & support.

Cons

- Lack of separation between external system and simulated humans probably impedes model development.
- Network representation has little advantage for complex systems.
- Psychological constraints have to be identified by the modeler and added by hand to each task in the network representation.
  - HOS micro-models and MHP values are included, but very limited.
  - May have to write special code to apply micro-models for each task.

MicroSAINT IPME/MS-HOS (continued)

Fragment of a really big network for Navy CIC model.


Initiate/Confirm Engage Order
Confirm Birds Away
Conduct Kill Evaluation
System Status
Track's visual display profile
Track is NOT covered
Track is covered (Y/N?)
Track is covered
In-bound missile msg
In-bound Missile (Y/N?)
Track's meeting ROEmsg
Track's meeting ROE (Y/N?)
Track's failure to respond to Level II Warning
Track's approaching its weapons range (Y/N?)
Track's approaching its weapons release range msg
Track's failure to respond to Illumination
Engage Orders Initiated
Track Verbal Report
'Issue Track Report' Msg
Engagement Order
'Initiate/Confirm Engagement' Msg
Initiate Engage Orders (Y/N?)
Engagement Solution
'Produce Engagement Solution' Msg
Engage Orders NOT Initiated
Firing Solution no longer a go
Firing Solution still a go Yes
Firing Solution still a go (Y/N?)
Release Missiles
Weapons Away (Y/N?)
Weapons Away Yes
'Weapons Firing' Msg
Kill Yes
Kill Evaluation (Y/N?)
'Kill Evaluation' Msg
System's moving tools releasing Weapons (Y/N?)
Weapons Away No
System's moving tools releasing Weapons
Engagement Status
System's failed in moving tools releasing Weapons
Review Engagement Status
INFORMATION Variables
DECISION Variables
ACTION Variables
OUTCOME Variables

COGNET/iGen

Url:

http://www.chisystems.com/

Originally developed to represent task analyses for training purposes.

Many applications described, including complex military tasks.

An extremely complex system.

Appears to include a black-board architecture, parallel execution of processes, a knowledge representation system, and many other mechanisms and subsystems, including recent additions of HOS and GOMS.

No public technical description appears to be available, limiting understanding and evaluation.

Scientific status, constraints represented, programming methodology, are unclear.

Pros

Works for very large and complicated tasks.

Commercial-level software support and training.

Cons

Apparent scientific isolation.

Summary Comparison

To what extent do the different architectures represent and enforce important constraints?

Well-grounded constraints - supported by developed theory and substantial empirical evidence, as opposed to speculative or arbitrary constraints.

Psychological - Human performance theory and data.

Representational - Computational experience and cognitive theory.

Constraint Comparison

	Constraints							
	Psychological						Representational	
	Visual	Auditory	Manual	Vocal	WM	Cognition	Declarative	Procedural
EPIC	++	+	++	++	!	++	++	++
EPIC-Soar	++	+	++	++	-	++		++
ACT-R 5, -R/PM	+	+	++	++	!	++	++	++
GLEAN	+	+	++	++	!	++	++	++
APEX/CPM-GOMS	+	+	+	+	?	+	?	+
MIDAS	+	?	+	+	?	-	-	-
IPME/MS-HOS	-	-	-	-	-	-		
COGNET	?	?	?	?	?	?	-	-

++ Good - constraints covered and enforced well
+ Some - gaps present in coverage/enforcement
- None - representable, but no enforcement
! Controversial representation and constraints
? Can’t tell from public sources
Does not appear to be represented

Summary

Several cognitive architecture systems were briefly summarized and compared.

Differ strongly in how well psychological constraints are represented and enforced.

Some are under active scientific development to incorporate additional constraints.

Others are popular and useful, but constraints are not well-represented.